

# FINAL PROJECT REPORT

## Resolve Now: Your Platform For Online Complaints

TeamID : LTVIP2025TMID20349

### INTRODUCTION

**ResolveNow** is an online complaint registration and management system designed to streamline the way individuals and organizations raise, track, and resolve grievances. The platform provides a centralized solution where users can easily register complaints, track their progress in real-time, and receive timely updates from agents or administrators.

Ordinary users can submit complaints through a user-friendly form and monitor their status. Agents are responsible for reviewing and addressing complaints assigned to them, with tools for updating complaint statuses and messaging users. Admins manage the overall platform—assigning complaints, managing users, and overseeing complaint resolution activity. By integrating efficient communication tools, structured workflows, and a secure backend, ResolveNow aims to enhance transparency, reduce resolution time, and increase user satisfaction in complaint handling processes.

### 1.1 Project Overview

**ResolveNow** is a full-stack web application developed to manage and streamline the process of handling customer complaints in a structured, efficient, and user-friendly manner. The system enables users to register complaints online, track their status, and communicate directly with agents handling their cases. Built using the MERN stack (MongoDB, Express.js, React.js, Node.js), ResolveNow leverages modern web technologies to offer real-time interactivity, secure data handling, and role-based accessibility.

The application supports three core user roles:

**Ordinary Users**, who can register, submit complaints, and monitor complaint progress.

**Agents**, who are assigned specific complaints and responsible for updating complaint status and resolving issues.

**Admins**, who oversee the system, manage user roles, assign complaints to agents, and handle escalations.

Key features include complaint assignment, message-based communication, real-time status tracking, and an admin dashboard for user and complaint management. With a responsive interface and scalable backend, ResolveNow is ideal for organizations aiming to improve their complaint resolution workflow and customer satisfaction.

### 1.2 Purpose

The primary purpose of *ResolveNow* is to provide a centralized, secure, and user-friendly platform for individuals and organizations to submit, track, and resolve complaints related to services or products. By digitizing and automating the complaint handling process, the platform improves transparency, responsiveness, and customer satisfaction. It acts as a bridge between users and customer service agents, facilitating smooth and accountable resolution workflows.

## 2. Ideation Phase

### 2.1 Problem Statement

Many organizations struggle with inefficient, delayed, and unorganized complaint handling processes that lead to user dissatisfaction and operational bottlenecks. Customers often lack a clear way to submit complaints, track their progress, or communicate with responsible personnel, while administrators face challenges in assigning, monitoring, and resolving issues in a timely manner. To address these gaps, there is a need for a centralized,

user-friendly platform that allows complaint registration, real-time tracking, role-based access, and smooth communication between users, agents, and administrators.

**ResolveNow** is developed to solve this problem by offering an online complaint management system that ensures transparency, accountability, and efficiency in complaint resolution workflows.

## 2.2 Empathy Map Canvas

### EAR

"I don't know if my complaint is even being looked at." — Users are often left wondering about the status of their submitted complaints.

"It's frustrating when I don't get any updates." — Lack of real-time notifications or acknowledgements causes uncertainty and dissatisfaction.

### SEE

Unclear interfaces or poorly structured forms when trying to submit a complaint.

No centralized view of all previous complaints or their current status.

Lack of transparency into which agent is handling the issue.

### SAY & DO

Call or email support teams repeatedly to follow up on complaints.

Try using social media or third-party platforms to escalate issues when not resolved.

Complain to peers about the slow or unhelpful support system.

### PAIN

Delays in resolution due to inefficient routing of complaints.

Frustration over having to explain the issue multiple times to different people.

Concerns over data privacy when sharing personal or sensitive complaint details.

### GAIN

A single, easy-to-use platform to register, track, and resolve complaints.

Clear communication with the assigned agent and timely updates on complaint status.

Improved trust in the resolution process due to transparency and structured workflows.

## 2.3 Brainstormed Features

**Role-Based Access:** Admin, Agent, and Ordinary User roles with customized dashboards and access restrictions.

**Complaint Approval and Assignment System:** Admin can view all incoming complaints and assign them to available agents based on expertise or workload.

**Complaint CRUD Operations:** Users can create complaints, track status updates; Admin and agents can update or close complaints.

**User Profile Management:** Admin can view, update, or delete user profiles; users can update their own details securely.

**Scalable Schema Design:** MongoDB schema supports extension with minimal rework, allowing future additions like categories, tags, etc.

**Agent Activity Panel:** Agents get a personalized view of complaints assigned to them with quick update options.

### 3. Requirement Analysis

The requirement analysis for the ResolveNow web application involves identifying the core user needs, system capabilities, and the technical architecture necessary to deliver a reliable, responsive, and secure platform. This analysis outlines both **functional** and **non-functional requirements** to ensure that the system meets expectations from user experience to backend performance and scalability.

#### 3.1 Customer Journey Map

Stage	Actions	Thoughts	Emotions	Opportunities for Improvement
1. Awareness	User sees a damaged product, poor service, or unresolved issue and searches for a way to file a complaint online.	"I need to report this quickly. Is there a platform I can trust?"	Frustrated, doubtful	SEO optimization, simple homepage with key benefits clearly listed.
2. Onboarding	Visits ResolveNow, browses homepage, then signs up with name, email, password, and selects user role.	"This looks legit. I hope it's secure and easy to use."	Curious, cautious	Smooth UI/UX, auto-validation in forms, optional Google sign-in.
3. Login	Logs in using registered credentials. Login persists via local storage.	"Let's hope I can access my dashboard easily."	Neutral to hopeful	Add token-based secure authentication with error feedback.
4. Complaint Submission	Navigates to dashboard and submits a complaint with necessary details, images, and address.	"Will someone actually look into this?"	Relieved yet uncertain	Add auto-suggestions, location API for easier input.
5. Tracking & Updates	Receives confirmation, tracks complaint status, gets notified when updates occur or when an agent is assigned.	"Great, they responded quickly. Let me check what's next."	Reassured, engaged	Push notifications or SMS alerts for offline users.
6. Agent Interaction	Receives a message from an agent. Chats via built-in messaging to clarify or explain complaint.	"It's helpful to speak directly with someone assigned."	Supported, confident	Enhance chat with typing indicators or file sharing.
7. Resolution	Gets final resolution—either replacement, refund, or service fix.	"I'm glad this was resolved without multiple follow-ups."	Satisfied, relieved	Add a "Resolution Summary" download or proof of action taken.
8. Return Visit	Logs back in weeks later to check complaint history or file a new one.	"I hope my data and history are still here."	Confident, familiar	Enable dashboard filters, export history features.

The journey of a user on the ResolveNow platform begins with **awareness**, typically triggered by a negative experience such as receiving a faulty product or unsatisfactory service. At this stage, the user actively looks for a trustworthy platform to register their complaint. Upon discovering ResolveNow, they visit the **homepage**, where clear navigation and an intuitive layout encourage them to sign up.

During the **onboarding phase**, the user fills in their details to create an account, selecting their role—Ordinary User, Agent, or Admin. This process should be smooth and secure, reassuring the user about the credibility of the platform. After registration, they proceed to the **login step**, which allows them to securely access their personalized dashboard, where they can manage or view complaints based on their role.

Once logged in, the user engages with the **complaint submission** feature. They enter complaint details such as the nature of the issue, location, supporting images, and any other relevant comments. The platform confirms the successful submission and initiates the tracking process. At this point, the user may still feel uncertain but is reassured by email or on-screen confirmation messages.

As the complaint moves through the system, the user enters the **tracking and notification** stage. Here, they receive real-time updates about the complaint's status, including whether it has been assigned, resolved, or requires more information. When an **agent is assigned**, direct communication is enabled through a built-in chat system, where users can clarify concerns and receive updates. This direct line builds user confidence and fosters a feeling of being heard and supported.

Once the issue is reviewed and resolved—either through a replacement, refund, or corrective service—the user receives a **resolution update**, concluding the primary interaction. This is often a satisfying stage, especially when it occurs swiftly and without multiple escalations. Following this, the user is encouraged to leave **feedback** on their experience with the platform and the agent. This feedback mechanism reinforces a sense of involvement and improves platform credibility.

In the final stage, **return visits**, users may log back in to check their complaint history or file new complaints. Having an accessible and well-organized dashboard ensures a positive long-term relationship with the platform. Overall, the ResolveNow platform transforms a potentially frustrating process into a structured, transparent, and user-centric experience.

### 3.2 Solution Requirements

The ResolveNow platform requires both functional and non-functional capabilities to meet user expectations and ensure system performance.

#### Functional Requirements

##### User Registration and Login

Users (Ordinary, Agent, Admin) must be able to register, log in, and access role-specific dashboards.

##### Role-Based Access Control (RBAC)

The system must provide separate views and privileges for each user type:

**Ordinary Users:** Submit and track complaints.

**Agents:** View and update assigned complaints.

**Admin:** Manage users, assign complaints, view system-wide analytics.

##### Complaint Management

Users can submit complaints including title, description, images, and location data.

Admin can assign complaints, and agents can update the status (e.g., Pending, Resolved, Cancelled).

##### Messaging System

Real-time messaging feature for users and agents to discuss the complaint.

##### User Management

Admin should be able to view, update, and delete users, as well as filter users by type.

##### Complaint History and Tracking

Users should have access to a list of all complaints they've submitted, along with real-time status updates.

#### Non-Functional Requirements

##### Usability

The interface should be intuitive and responsive for all user types, using modern UI frameworks like React.js, Material UI, and Bootstrap.

### Performance

The system should respond quickly to user actions, especially for complaint status updates and data retrieval.

### Scalability

Backend architecture should support growing numbers of users, agents, and complaints without performance degradation.

### Security

User data and complaints should be stored securely using Mongoose + MongoDB. Login authentication must be implemented with secure password handling.

### Maintainability

Modular code structure using React.js for frontend and Express.js for backend ensures ease of updates and feature expansion.

### Data Integrity

Complaint and user data should be validated and sanitized both client- and server-side to ensure accuracy and prevent injection attacks.

## 3.3 Data Flow Diagram

A **Data Flow Diagram (DFD)** illustrates how data moves within the Resolve Now platform. It captures how users interact with the system, how information flows between different components, and where the data is stored.



A Data Flow Diagram (DFD) is a graphical representation of the flow of data within a system. It helps visualize how information is input, processed, and output across different modules of the application. In the context of the **ResolveNow** platform, the DFD illustrates how data moves between users (Ordinary Users, Agents, and Admins) and the backend system (Node.js + MongoDB) for complaint registration, assignment, tracking, and resolution.

### Level 0 DFD – Context Diagram

The **Level 0 DFD**, also known as the **context-level diagram**, provides a high-level overview of the **ResolveNow** system as a single unified process. It outlines how various external user entities interact with the system and what kind of data is exchanged.

**Ordinary Users** interact with the system to **register**, **log in**, **submit complaints**, and **track the status** of their complaints.

**Agents** log in to the system, **receive assigned complaints**, **update statuses**, and **communicate** with users.

**Admins** perform **user approvals**, **assign complaints** to agents, **view system-wide activity**, and manage user or complaint records.

The system relies on a **MongoDB database** to store and retrieve information like user credentials, complaints, messages, and assignments.

This level does not show internal operations but defines the major **input/output boundaries** of the platform.

## Level 1 DFD – Process Breakdown

The **Level 1 DFD** breaks down the main ResolveNow system into **distinct modules**, describing how each unit processes the data it receives and how it interacts with the database.

### User Authentication Module

Manages **registration** and **login** for all user types (Admin, Agent, Ordinary).

Accepts credentials, performs validation, and returns the **user object** for session handling.

User details (like name, email, password, role) are stored securely in the database.

Currently, the project uses **localStorage** to maintain session state (not JWT or sessions).

### Complaint Registration Module

Allows users to **submit new complaints**.

Accepts data like complaint type, address, description, and stores it in the **Complaints collection**.

Links each complaint to the corresponding **userId** for tracking.

### Complaint Assignment Module

Used by **Admins** to assign complaints to Agents.

Captures agentId and complaintId, then stores them in the **AssignedComplaints collection**.

Prevents assignment of complaints with a "Cancelled" status.

### Complaint Resolution & Tracking Module

Allows **Agents** to update the **status** of assigned complaints (e.g., Resolved, In Progress).

Allows **Users** to view complaint status through the **Status page**.

Real-time updates and feedback options are available.

### Messaging and Communication Module

Enables messaging between **Users and Agents** using the /messages endpoint.

Messages are stored in the **Message collection**, filtered by complaintId.

### Admin Control Module

Grants **Admins** full access to all **user and complaint records**.

Enables user verification, complaint assignment, and **deletion or update** operations.

Provides a central dashboard for platform governance.

### Notification & Feedback Module

Sends **confirmation messages** after complaint submission or resolution.

Users can **submit feedback** or update their **profile information**.

Admins and Agents get **real-time updates** on complaint status and actions.

### Data Stores

The ResolveNow backend interacts with a **MongoDB database**, with key collections:

**Users** – Contains user credentials and roles (Admin, Agent, Ordinary).

**Complaints** – Stores all complaint records linked to userId.

**AssignedComplaints** – Tracks which agent is assigned to which complaint.

**Messages** – Stores chat messages between users and agents.

All collections are managed using **Mongoose**, which ensures schema validation and seamless database operations.

### External Entities

**Ordinary User:** Registers, submits complaints, receives updates, chats with agents.

**Agent:** Logs in, views assigned complaints, updates statuses, chats with users.

**Admin:** Manages user approvals, assigns complaints, oversees all activity.

All communication between the front-end (React) and back-end (Express.js) is done through **RESTful APIs** over **HTTP**, with request handling, validation, and error management performed on the server.

## 3.4 Technology Stack

The **ResolveNow** application is built using the **MERN stack**, supplemented by various tools and libraries to create a seamless, efficient, and user-friendly complaint registration and tracking system.

### Frontend Technologies:

**React.js** is used to develop a dynamic and component-based user interface.

**React Router** manages client-side routing, enabling smooth navigation between login, dashboard, complaint form, and admin/agent panels.

**Bootstrap** and **Material UI** are employed for responsive design and consistent styling.

**Axios** is used for making HTTP requests to the backend API for actions like user login, complaint registration, and fetching complaint statuses.

### Backend Technologies:

**Node.js** provides the server-side runtime environment.

**Express.js** is used to define API endpoints, manage routing, and handle middleware logic.

**CORS** and **body-parser (via express.json())** are used for API security and data parsing.

**Multer** is used for future enhancement potential (e.g., uploading documents or image evidence for complaints).

#### Database:

**MongoDB** is used to store all structured data including user records, complaint entries, assignment records, and chat messages.

**Mongoose** (ODM) simplifies MongoDB schema modeling and provides built-in validation and query capabilities.

#### Authentication & Security:

Currently, **localStorage** is used to persist login sessions by storing the user object and role.

Plans for future upgrades include adding **JWT (JSON Web Tokens)** for secure stateless authentication and **bcryptjs** for hashing user passwords.

#### Other Tools and Libraries:

**Nodemon** improves developer productivity by auto-restarting the backend server on file changes.

**Git** and **GitHub** are used for version control and code collaboration.

**Postman** assists in testing API endpoints during development phases.

**VS Code** serves as the primary code editor for frontend and backend development.

## 4. Project Design

### 4.1 Problem Solution Fit

The Problem–Solution Fit ensures that the *ResolveNow* platform effectively addresses the inefficiencies in traditional complaint handling systems and meets the real-world needs of both citizens and administrative authorities. This validation is crucial before scaling the system across departments or regions.

This project ensures a strong problem–solution fit by directly addressing those gaps through:

- Create a transparent, accountable channel for citizens and organizations to report complaints or issues.
- Centralize complaint management—submission, tracking, resolution—on one platform.
- Provide real-time status updates and automated notifications for better engagement.
- Empower administrators with analytics and case tracking for better governance.
- Build trust through timely responses, escalation mechanisms, and feedback collection.

The solution ensures that both civic engagement needs and technical scalability are met while creating a streamlined and trustworthy complaint resolution experience.



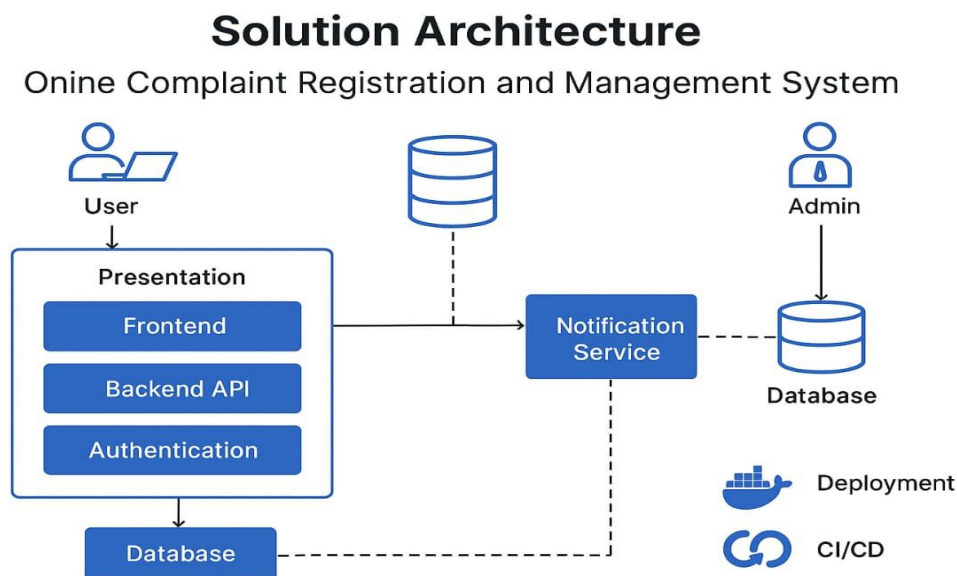
## 4.2 Proposed Solution

The Resolve Now platform is a full-stack web application (React frontend + Node.js backend) that enables clients to post jobs, freelancers to apply, and both parties to collaborate through real-time messaging and secure payments. Admin controls help manage disputes and community standards.

**ResolveNow**, a full-stack complaint registration and management system, offers:

- Online complaint submission with unique tracking ID
- Role-based dashboards for complainants, staff, and admins
- Real-time status updates, email/SMS alerts, and escalation paths
- Complaint categorization, prioritization, and automated assignment
- Performance metrics and reports for timely resolution tracking

## 4.3 Solution Architecture



The solution architecture for **Resolve Now** ensures a robust, user-friendly, and scalable platform that connects clients with skilled freelancers. The architecture focuses on responsive design, secure transactions, intelligent matching, and real-time messaging to ensure smooth project workflows and high user satisfaction.

- Seamless Crime posting on the application
- End-to-end project lifecycle management
- Scalable user authentication and authorization
- Secure and trackable payment transactions

The architecture of the ResolveNow application adopts a modular client-server model, clearly separating concerns across the frontend, backend, and database layers.

### Frontend (Client-Side)

The frontend is developed using **React.js**, which provides a dynamic and responsive interface for different user roles—Admin, Agent, and Ordinary users. It manages all user interactions such as login, complaint registration,

status tracking, and administrative dashboards. **Axios** is used for sending and receiving data to and from the backend via **RESTful APIs**. Routing within the application is handled using **React Router**, with protected routes implemented based on the user role stored in local storage. The UI is enhanced using **Bootstrap** and **Material UI**, ensuring a clean and consistent experience across devices.

**Backend (Server-Side)**

The backend is built on **Node.js** using the **Express.js** framework, which handles all the business logic, routing, and API requests. It manages tasks such as user authentication, complaint submission, complaint assignment to agents, and real-time chat/message handling. While the current implementation uses **local storage** to manage login sessions, future enhancements are planned to incorporate **JWT-based authentication** and **bcryptjs** for password encryption. Middleware support, error handling, and route grouping ensure a maintainable and scalable server architecture.

**Database (MongoDB)**

**MongoDB** is used as the database to store all persistent application data such as user profiles, complaints, messages, and assigned complaints. **Mongoose**, a powerful ODM (Object Document Mapper), is used to define schemas and perform data validation, querying, and relationships between documents. Collections such as Users, Complaints, AssignedComplaints, and Messages represent key entities in the system.

**Data Flow and Communication**

The React frontend communicates with the Express backend through HTTP requests. The backend validates and processes these requests, performs operations using Mongoose to interact with the MongoDB database, and returns responses in JSON format. Although token-based security is not yet implemented, access control is enforced on the client side using user roles stored in the browser. Future upgrades will include JWT tokens for session security and improved backend authorization middleware.

This architecture supports real-time performance, modular development, and future scalability, ensuring that the ResolveNow system can grow with increasing users and expanding feature sets.

**5. Project Planning And Scheduling**

**5.1Project Planning**

The **ResolveNow** project follows the **Agile Scrum methodology**, an iterative and incremental approach to software development that promotes flexibility, collaboration, and rapid delivery. The development process is divided into short, manageable sprints, each typically lasting 1-2 weeks, with defined goals and deliverables. This allows the team to focus on building core features like user registration, complaint submission, assignment workflows, and admin dashboards in phased cycles.

**Product Backlog & Sprint Schedule**

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priorit y	Team Members
Sprint-1	User Authentication	USN-1	As a user, I can sign up and log in securely.	3	High	Manntasha,Priyanka
Sprint-2	Complaint	USN-3	As a user, I can submit a complaint	2	High	Varun

	<b>Submission</b>		with relevant details and documents.			
<b>Sprint-3</b>	<b>Complaint Tracking &amp; Communication</b>	USN-4	As an agent, I can view and manage complaints assigned to me.	3	High	Varun,Vamsi
		USN-5	As a user, I can track the status of my complaint in real-time	2	High	Priyanka
		USN-6	As a user, I can chat with the assigned agent regarding my complaint.	2	Medium	Manntash
<b>Sprint-4</b>	<b>Complaint cancellation and integration of chatbot</b>	USN-7	As a user, I can chat with chatbot regarding issues of the complaints	3	High	Manntasha,Varun
		USN-8	As a user, I can cancel the complaint if the issue resolved	2	Medium	Vamsi
		USN-9	As a user, I can talk with chatbot regarding the complaint cancellation and update status	2	Medium	Varun

The development cycle for the *ResolveNow* application was structured into **four sprints**, each aligned with a specific set of functional requirements (epics) and user stories. This sprint-based approach allowed the team to incrementally deliver key features while maintaining clarity and efficiency throughout the development process.

Sprint-1 focused on implementing the core user authentication module, enabling secure login and registration for users.

Sprint-2 addressed the complaint submission module, allowing users to lodge complaints with necessary details and file uploads.

Sprint-3 centered on complaint tracking and communication features. Agents were enabled to view and manage their assigned complaints, while users could monitor complaint status in real time and chat with agents, enhancing transparency and interaction.

Sprint-4 completed the system with new key functionalities such as **complaint cancellation by users** and the **integration of a chatbot**. The complaint cancellation feature allowed users to withdraw complaints if no longer relevant, updating the complaint status and notifying the assigned agent. Simultaneously, a rule-based chatbot was embedded using IBM Watsonx Assistant, enabling users to get instant support for common queries like complaint status, registration help, or contact information.

Each sprint was executed over six days with a story point target of 20 points. Tasks were prioritized based on business value and technical complexity. Sprint progress was tracked using burndown charts and sprint trackers that highlighted planned vs. actual delivery. Additional functionalities such as owner access approval, admin dashboards, role-based routing, property image uploads, and filter-based searching were included in the backlog and distributed across sprints. This structured and iterative approach helped the team maintain clear goals, adaptive development cycles, and consistent productivity.

## 6. Functional and Performance Testing

### 6.1 Performance Testing

**ResolveNow** is an efficient and user-friendly complaint registration and management platform that empowers citizens to report issues, track progress, and communicate with responsible agents, while allowing administrators to oversee and assign complaints. Key features include:

- Role-based access for Admins, Agents, and Users
- Complaint submission with detailed issue information
- Real-time status tracking of complaints
- Agent dashboard for viewing and updating assigned complaints
- In-app chat functionality between users and agents
- Admin portal for managing users, agents, and complaints
- Complaint cancellation by users
- Integration with rule-based chatbot for instant support and FAQs
- Secure user authentication with role-specific access

Testing Period: 06-06-2025 to 07-06-2025

#### Testing Scope

- User registration and login
- Complaint Submission
- Status Tracking
- Agent Communication
- Complaint Assignment by Admin
- Admin Dashboard Functionality
- Responsive User Interface (UI)
- Logout and Session Management

#### Requirements to be Tested:

- As a Client of ResolveNow, I want to Post the complaints easily.
- As a user, I want secure login and Post complaints.
- As an admin, I want to manage reported users and disputes.

#### Testing Environment

- URL: <https://reslovenow.example.com>

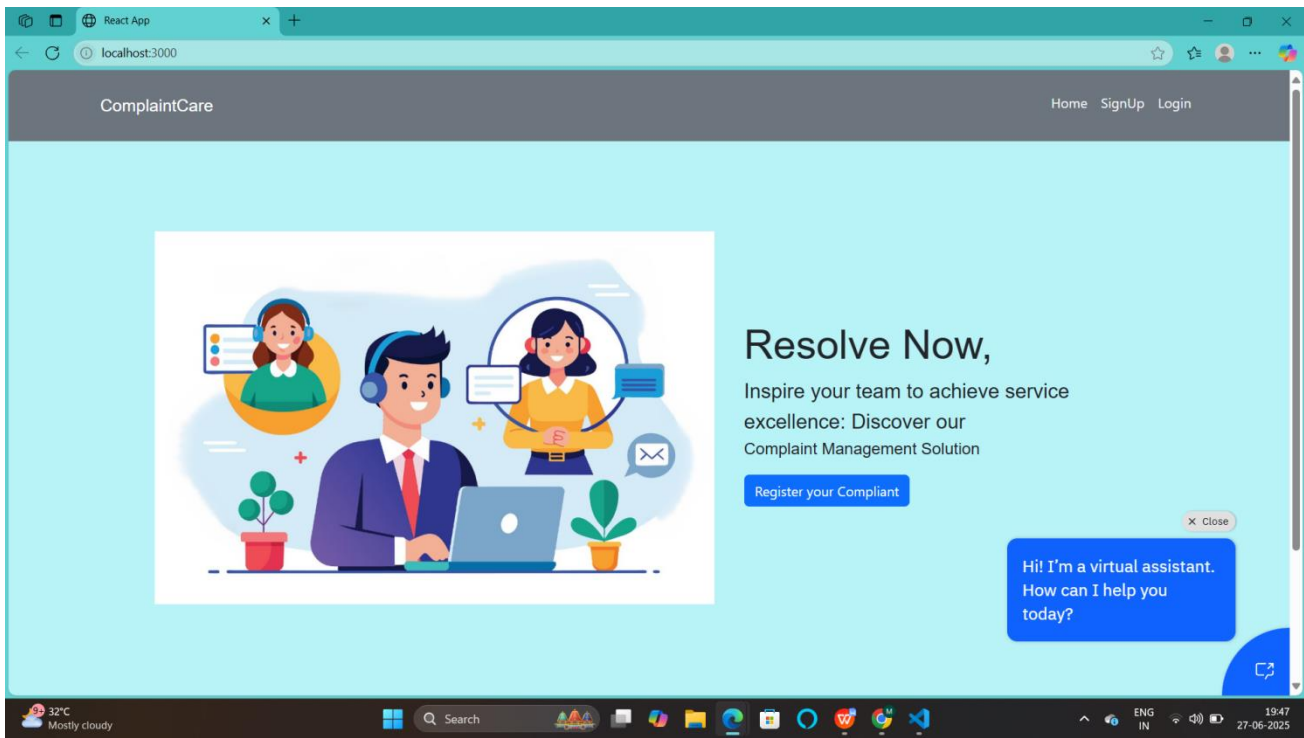
#### Test Cases:

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Pass/Fail
TC-001	User Registration & Login	1. Visit site 2. Click "Sign Up" 3. Fill & submit form	User account is created and redirected to dashboard	Account created, redirected to dashboard	Pass
TC-002	Submit Complaint Without Details	1. Login as user 2. Go to "Submit Complaint" 3. Leave all fields empty 4. Click Submit	System should show validation error messages	Form gets submitted without showing any warning	Fail
TC-003	Agent Communication with User	1. Login as agent 2. Open assigned complaint 3. Message the user	User should receive message and be able to reply	Messages exchanged between user and agent	Pass
TC-004	Login with wrong password	1. Go to login page 2. Enter valid email but incorrect password 3. Click "Login"	Error message like "Invalid credentials" shown	Login page reloads without message	Fail (if no message shown) or Pass if handled
TC-005	UI Responsiveness (Mobile)	1. Open system on phone browser 2. Navigate through login, dashboard, forms	All pages should display correctly and be usable on mobile	Layout adapts correctly on all screen sizes	Pass

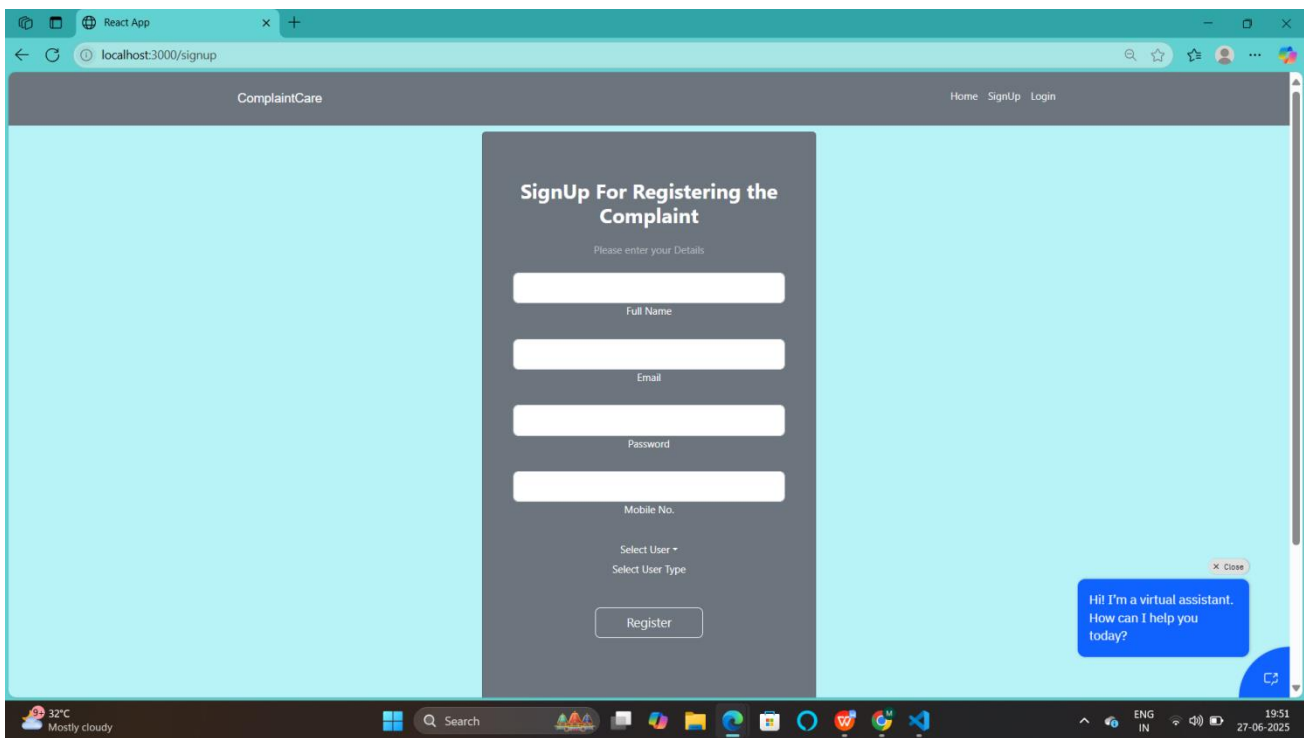
## 7. Results

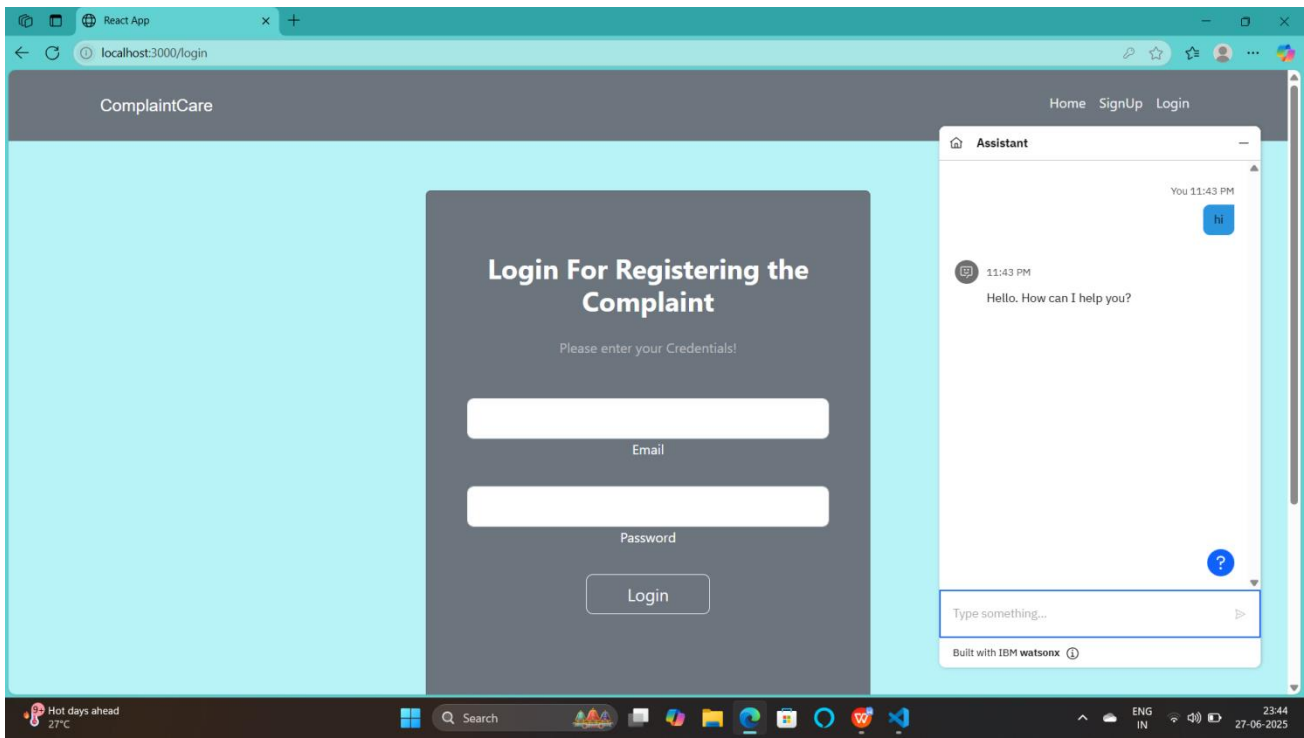
### 7.1 Output ScreenShots

- Landing Page

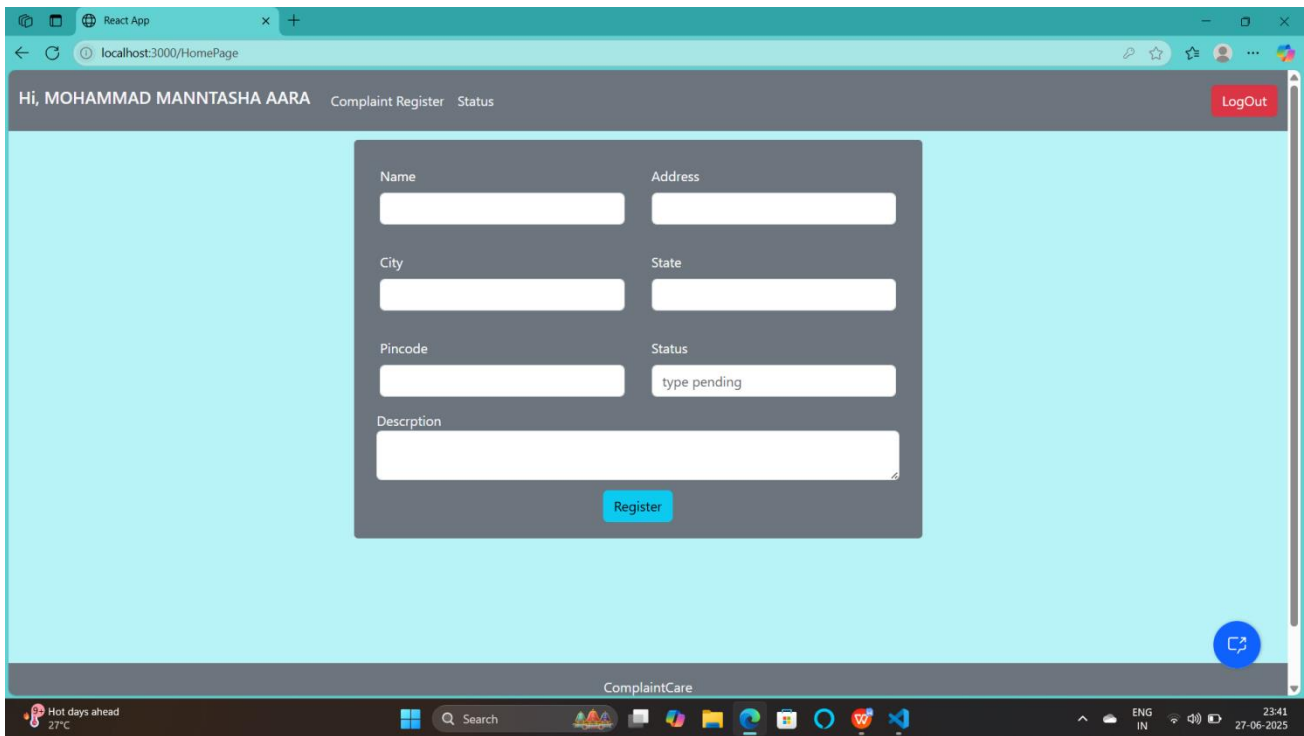


## ● Registration and Login Page

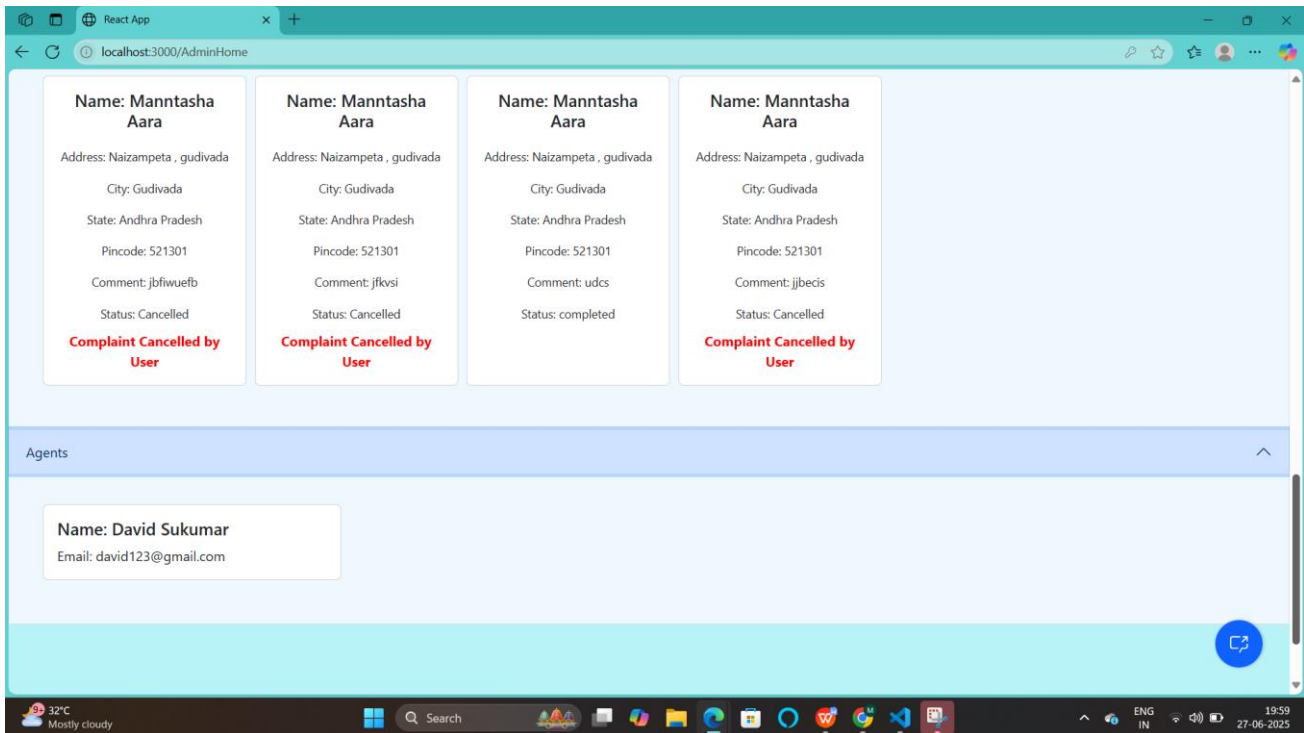




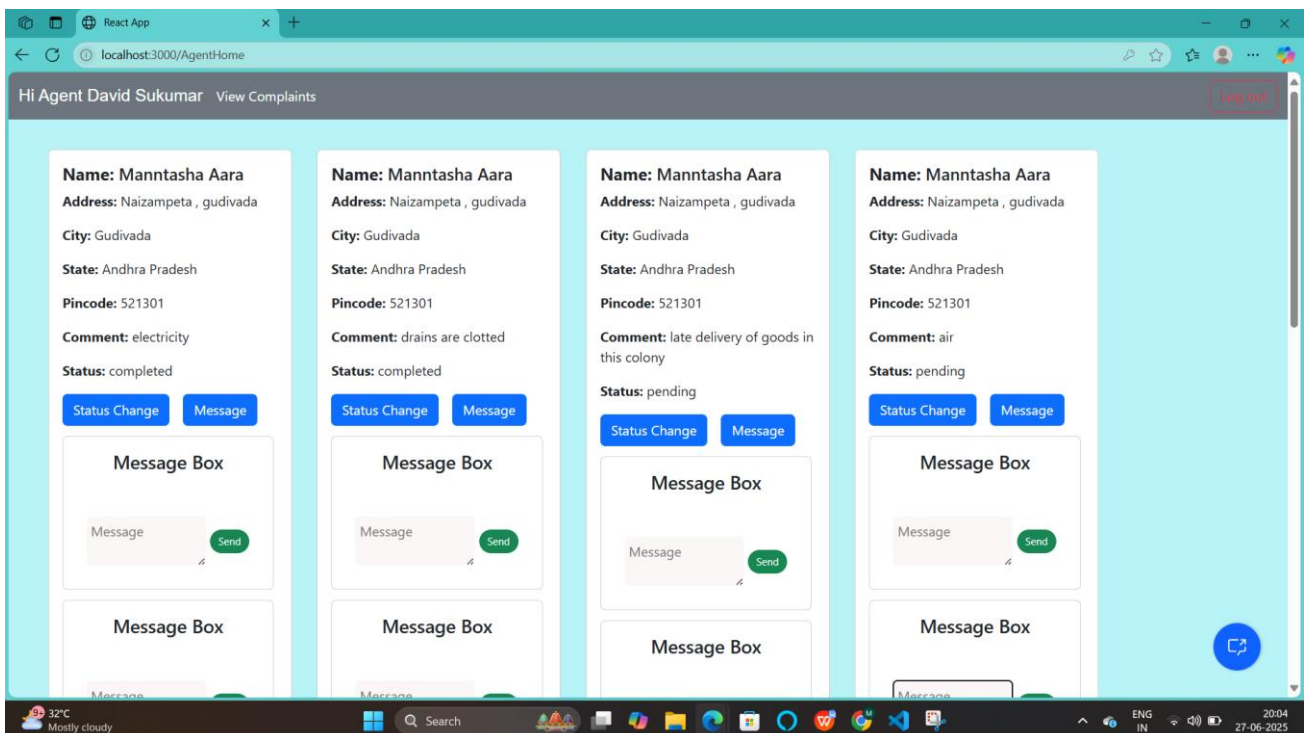
- Home Page
- User



- Admin



- Agent



## 8. Advantages and Disadvantages

### Advantages

#### Efficient Complaint Handling

Users can register complaints quickly, and the system ensures they are assigned to the appropriate agent, reducing manual workload and delays.

#### Role-Based Access Control

Separate views and functionalities for Admins, Agents, and Users enhance security and simplify navigation and access.



### **Real-Time Complaint Tracking**

Users can monitor the status of their complaints, increasing transparency and user satisfaction.

### **Chat Integration**

In-app messaging between users and agents facilitates smooth communication and quicker issue resolution.

### **Admin Oversight**

Admins can manage users, review complaints, assign agents, and delete inappropriate data, offering complete control over the system.

### **Complaint Cancellation Feature**

Users can cancel a complaint if no longer needed, preventing unnecessary work for agents and ensuring database accuracy.

### **Disadvantages**

#### **Lack of Advanced Authentication**

Currently, it lacks token-based authentication (e.g., JWT), which may reduce security in case of session hijacking or unauthorized access.

#### **No Real-Time Notifications**

The platform does not support push or email notifications for complaint updates, which may lead to missed updates.

#### **Limited Chat Functionality**

The messaging system is basic and does not support features like file sharing, typing indicators, or read receipts.

#### **No Offline Support**

The application depends entirely on an internet connection and cannot queue actions for later sync.

#### **Admin Dependency**

Complaint assignment relies solely on admin input; no automation or AI-based routing exists.

#### **Scalability Constraints in UI**

As the number of complaints and users grows, the UI might need pagination, filtering, and optimization for performance.

## **9. Conclusion**

The **ResolveNow** complaint registration and management system provides an efficient, user-friendly platform for handling service-related grievances. By offering a structured and role-based solution for users, agents, and administrators, it ensures that complaints are lodged, tracked, and resolved in a timely manner. The integration of key features like real-time complaint tracking, chat-based communication, and administrative control streamlines the entire workflow from complaint submission to resolution. Built on the MERN stack, the application is scalable, maintainable, and aligns with modern web development standards. The successful inclusion of complaint cancellation and chatbot features further enhances user experience and operational ease.

## **10. Future Scope**

### **JWT-Based Authentication & Authorization:**

Implementing secure token-based login will protect user sessions and enable scalable security practices.

### **Push & Email Notifications:**

Real-time alerts for complaint updates, assignments, and messages can improve user engagement and responsiveness.

### **AI-Powered Chatbot:**

Enhancing the chatbot to use AI/NLP will enable it to handle a broader range of user queries and automate support tasks.

**Complaint Auto-Assignment:**

Introducing logic to automatically assign complaints based on agent availability, category, or location will optimize workload distribution.

**Data Analytics Dashboard:**

Admins could benefit from analytics showing complaint trends, agent performance, and resolution rates for better decision-making.

**Mobile App Development:**

Creating a mobile version of ResolveNow will make it more accessible to users and agents on the go.

**Multilingual Support:**

Adding support for multiple languages can expand usability across regions and enhance inclusivity.

**Role-Based Notification Preferences:**

Allowing users and agents to customize when and how they receive updates would improve usability.

These improvements will make **ResolveNow** a robust, scalable platform capable of serving both small organizations and large institutions seeking efficient complaint management systems.

## 11. Appendix

**Source Code**

[https://drive.google.com/drive/folders/15iSWaQQ0D6lb5iUAZ\\_fl6FaDTAP0LeYw?usp=sharing](https://drive.google.com/drive/folders/15iSWaQQ0D6lb5iUAZ_fl6FaDTAP0LeYw?usp=sharing)

**Poject Demo Link**

<https://drive.google.com/drive/folders/16PC0QjRFNOi3znjhStWSitIon5gzAnR2?usp=sharing>